I 6

```
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNNNNN      NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN   NNN   NNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN      NNNNNN  VVV         VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
CCC              OOO         OOO  NNN         NNN     VVV   VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
   CCCCCCCCCCCC     OOOOOOOOO     NNN         NNN        VVV
```

FILEID**CONVSORT

```
  CCCCCCCC    OOOOOO    NN      NN  VV      VV   SSSSSSSS    OOOOOO    RRRRRRRR    TTTTTTTTTT
  CCCCCCCC    OOOOOO    NN      NN  VV      VV   SSSSSSSS    OOOOOO    RRRRRRRR    TTTTTTTTTT
CC          OO      OO  NN      NN  VV      VV  SS          OO      OO  RR      RR      TT
CC          OO      OO  NN      NN  VV      VV  SS          OO      OO  RR      RR      TT
CC          OO      OO  NNNN    NN  VV      VV  SS          OO      OO  RR      RR      TT
CC          OO      OO  NNNN    NN  VV      VV  SS          OO      OO  RR      RR      TT
CC          OO      OO  NN  NN  NN  VV      VV   SSSSSS     OO      OO  RRRRRRRR      TT
CC          OO      OO  NN  NN  NN  VV      VV   SSSSSS     OO      OO  RRRRRRRR      TT
CC          OO      OO  NN    NNNN  VV      VV         SS   OO      OO  RR  RR        TT
CC          OO      OO  NN    NNNN  VV      VV         SS   OO      OO  RR  RR        TT
CC          OO      OO  NN      NN    VV  VV           SS   OO      OO  RR    RR      TT
CC          OO      OO  NN      NN    VV  VV           SS   OO      OO  RR    RR      TT
  CCCCCCCC    OOOOOO    NN      NN      VV       SSSSSSSS    OOOOOO    RR      RR      TT
  CCCCCCCC    OOOOOO    NN      NN      VV       SSSSSSSS    OOOOOO    RR      RR      TT

LL            IIIIII      SSSSSSSS
LL            IIIIII    SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;    1          0001  0 %TITLE  'VAX-11 CONVERT'
;    2          0002  0 MODULE  CONV$SORT       ( IDENT='V04-000',
;    3          0003  0                         OPTLEVEL=3
;    4          0004  0                         ) =
;    5          0005  0
;    6          0006  1 BEGIN
;    7          0007  1
;    8          0008  1 !**************************************************************************
;    9          0009  1 !*                                                                        *
;   10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;   11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
;   12          0012  1 !*  ALL RIGHTS RESERVED.                                                 *
;   13          0013  1 !*                                                                        *
;   14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
;   15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
;   16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
;   17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
;   18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
;   19          0019  1 !*  TRANSFERRED.                                                          *
;   20          0020  1 !*                                                                        *
;   21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
;   22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
;   23          0023  1 !*  CORPORATION.                                                          *
;   24          0024  1 !*                                                                        *
;   25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
;   26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.               *
;   27          0027  1 !*                                                                        *
;   28          0028  1 !*                                                                        *
;   29          0029  1 !**************************************************************************
```

```
;    31          0030  1 !++
;    32          0031  1 !
;    33          0032  1 ! Facility:     VAX-11 CONVERT
;    34          0033  1 !
;    35          0034  1 ! Abstract:     CONVERT routines wich sort the input file on the output
;    36          0035  1 !               files primary key and to sort the output file by it's
;    37          0036  1 !               secondary key
;    38          0037  1 !
;    39          0038  1 ! Contents:
;    40          0039  1 !               SORT_PRIMARY
;    41          0040  1 !               SORT_SECONDARY
;    42          0041  1 !               SET_OP_SORT
;    43          0042  1 !
;    44          0043  1 ! Environment:
;    45          0044  1 !
;    46          0045  1 !               VAX/VMS Operating System
;    47          0046  1 !
;    48          0047  1 !--
;    49          0048  1
;    50          0049  1 !
;    51          0050  1 ! Author:       Keith B Thompson        Creation date:  August-1980
;    52          0051  1 !
;    53          0052  1 !
;    54          0053  1 ! Modified by:
;    55          0054  1 !
;    56          0055  1 !       V03-013 RAS0272         Ron Schaefer            16-Mar-1984
;    57          0056  1 !               Allow CONVERT to fastload/sort network files, now that
;    58          0057  1 !               SORT-32 can handle them.
;    59          0058  1 !
;    60          0059  1 !       V03-012 RAS0260         Ron Schaefer            6-Mar-1984
;    61          0060  1 !               Modify input file specs for SORT for LIB$FIND_FILE.
;    62          0061  1 !
;    63          0062  1 !       V03-011 KBT0502         Keith B. Thompson       19-Apr-1983
;    64          0063  1 !               Remove reference to SOR$M_SIGNAL
;    65          0064  1 !
;    66          0065  1 !       V03-010 KBT0467         Keith B. Thompson       21-Jan-1983
;    67          0066  1 !               Don't bother calling set_key_desc in sort_primary because
;    68          0067  1 !               we don't know if the file is still open for block io and
;    69          0068  1 !               set_key_desc does a $read.  Also use the new sort interface.
;    70          0069  1 !
;    71          0070  1 !       V03-009 KBT0426         Keith B. Thompson       30-Nov-1982
;    72          0071  1 !               Fix a naming problem with the convert termorary file
;    73          0072  1 !               and remove sort error routine to get ready for the new
;    74          0073  1 !               sort interface which will signal errors.
;    75          0074  1 !
;    76          0075  1 !       V03-008 KBT0393         Keith B. Thompson       29-Oct-1982
;    77          0076  1 !               Use new set_key_desc routine
;    78          0077  1 !
;    79          0078  1 !       V03-007 KBT0379         Keith B. Thompson       21-Oct-1982
;    80          0079  1 !               Fix the linkage definition to set_key_block
;    81          0080  1 !
;    82          0081  1 !       V03-006 KBT0348         Keith B. Thompson       4-Oct-1982
;    83          0082  1 !               Use new linkage definitions (and fix history error
;    84          0083  1 !               in cwh0001!)
;    85          0084  1 !
;    86          0085  1 !       V03-005 CWH0001         CW Hobbs                17-Aug-1982
;    87          0086  1 !               Fix a history error in the last packet.
```

```
;    88          0087  1 !
;    89          0088  1 !       V03-004 KBT0125         Keith B. Thompson       10-Aug-1982
;    90          0089  1 !               Get the file name length from RSL not RSS
;    91          0090  1 !
;    92          0091  1 !       V03-003 KBT0045         Keith Thompson          9-Apr-1982
;    93          0092  1 !               Correct the way packed decimal sizes are given to sort
;    94          0093  1 !               Also fix when we do stable sorts ie. only with dups
;    95          0094  1 !
;    96          0095  1 !       V03-002 KBT0027         Keith Thompson          30-Mar-1982
;    97          0096  1 !               Chain the sort error messages
;    98          0097  1 !
;    99          0098  1 !       V03-001 KBT0014         Keith Thompson          17-Mar-1982
;   100          0099  1 !               Pass sort a lrl so it will not choke on sys$input
;   101          0100  1 !
;   102          0101  1 !****
```

```
;  104          0102  1
;  105          0103  1 PSECT
;  106          0104  1         OWN     = _CONV$OWN     (PIC),
;  107          0105  1         GLOBAL  = _CONV$GLOBAL  (PIC),
;  108          0106  1         PLIT    = _CONV$PLIT    (SHARE,PIC),
;  109          0107  1         CODE    = _CONV$CODE    (SHARE,PIC);
;  110          0108  1
;  111          0109  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  112          0110  1 LIBRARY 'SRC$:CONVERT';
;  113          0111  1
;  114          0112  1 FORWARD ROUTINE
;  115          0113  1         CONV$$SORT_PRIMARY      : CL$SORT_PRIMARY,
;  116          0114  1         CONV$$SORT_SECONDARY    : CL$SORT_SECONDARY,
;  117          0115  1         SET_UP_SORT             : CL$JSB_REG_11 NOVALUE;
;  118          0116  1
;  119          0117  1 DEFINE_ERROR_CODES;
;  120          0118  1
;  121          0119  1 EXTERNAL ROUTINE
;  122          0120  1         CONV$$GET_VM            : CL$GET_VM,
;  123          0121  1         CONV$$OPEN_IN,
;  124          0122  1         CONV$$RMS_OPEN_ERROR,
;  125          0123  1         CONV$$SET_KEY_DESC      : CL$SET_KEY_DESC,
;  126          0124  1         CONV$$SEARCH_FILE,
;  127          0125  1         LIB$PUT_OUTPUT          : ADDRESSING_MODE(GENERAL),
;  128          0126  1         SOR$BEGIN_SORT          : ADDRESSING_MODE(GENERAL),
;  129          0127  1         SOR$PASS_FILES          : ADDRESSING_MODE(GENERAL),
;  130          0128  1         SOR$SORT_MERGE          : ADDRESSING_MODE(GENERAL),
;  131          0129  1         SOR$END_SORT            : ADDRESSING_MODE(GENERAL);
;  132          0130  1
;  133          0131  1 EXTERNAL
;  134          0132  1         CONV$GL_SORT            : LONG,
;  135          0133  1         CONV$GL_WORK_F          : LONG,
;  136          0134  1
;  137          0135  1         CONV$AB_FLAGS           : BLOCK [ ,BYTE ],
;  138          0136  1
;  139          0137  1         CONV$AR_OUT_FILE_NAM    : REF DESC_BLK,         ! Output File
;  140          0138  1         CONV$GB_CURRENT_FILE    : BYTE,
;  141          0139  1         CONV$GL_FILE_COUNT,
;  142          0140  1         CONV$AR_PROLOGUE,
;  143          0141  1         CONV$GW_MAX_REC_SIZ     : WORD,
;  144          0142  1
;  145          0143  1         CONV$AB_IN_NAM          : $NAM_DECL,
;  146          0144  1         CONV$AB_IN_FAB          : $FAB_DECL,
;  147          0145  1         CONV$AB_IN_RAB          : $RAB_DECL,
;  148          0146  1         CONV$AB_OUT_NAM         : $NAM_DECL,
;  149          0147  1         CONV$AB_OUT_FAB         : $FAB_DECL,
;  150          0148  1         CONV$AB_OUT_RAB         : $RAB_DECL;
;  151          0149  1
;  152          0150  1 EXTERNAL LITERAL
;  153          0151  1         SOR$M_STABLE,
;  154          0152  1         SOR$GK_RECORD,
;  155          0153  1         SOR$GK_ADDRESS,
;  156          0154  1         SOR$GK_INDEX;
;  157          0155  1
;  158          0156  1 !       SORT Temporary File Name Data
;  159          0157  1 !
;  160          0158  1 BIND
```

```
;  161     0159  1          CONV_TMP_STR    = UPLIT ('CONVWORK'),              ! Convert Temp. File Name
;  162     0160  1          CONV_DEF_STR    = UPLIT ('SYS$SCRATCH:.TMP');      ! Default name
;  163     0161  1
;  164     0162  1 LITERAL
;  165     0163  1          CONV_TMP_SIZ = 8,
;  166     0164  1          CONV_DEF_SIZ = 16;
;  167     0165  1
;  168     0166  1 OWN
;  169     0167  1          CONV_TMP_DESC   : DESC_BLK,              ! Convert temp. file desc.
;  170     0168  1          TEMP_DESC       : DESC_BLK,              ! Expanded input file desc
;  171     0169  1
;  172     0170  1          ! Name block
;  173     0171  1          !
;  174     0172  1          RFA_NAM         : $NAM_DECL,             ! RFA Name Block
;  175     0173  1
;  176     0174  1          ! The fop bits are: Truncate eof - so to shrink file on multiple sorts
;  177     0175  1          !                   Defered write - of course
;  178     0176  1          !                   Create if - We know sort is doing a create but we
;  179     0177  1          !                               have created the file for him
;  180     0178  1          !
;  181     0179  1          FOP             : LONG INITIAL( FAB$M_TEF+FAB$M_DFW+FAB$M_CIF ),
;  182     0180  1          FILETYPE        : BYTE,
;  183     0181  1          RECORDFMT       : BYTE,
;  184     0182  1          RECORDSIZ       : WORD;
;  185     0183  1
;  186     0184  1 GLOBAL
;  187     0185  1
;  188     0186  1          CONV$GL_RFA_BUFFER      : LONG,                  ! Pointer to RFA Buffer
;  189     0187  1
;  190     0188  1          ! Work Files
;  191     0189  1          !
;  192     0190  1          CONV$AB_RFA_FAB         : $FAB_DECL,             ! RFA File FAB
;  193     0191  1
;  194     0192  1          CONV$AB_RFA_RAB         : $RAB_DECL;             ! RFA File RAB
;  195     0193  1
```

```
;   197     0194  1 %SBTTL  'INIT_SORT'
;   198     0195  1 ROUTINE INIT_SORT : NOVALUE =
;   199     0196  1 !++
;   200     0197  1 !
;   201     0198  1 ! Functional Description:
;   202     0199  1 !
;   203     0200  1 !       Initializes the rfa rms blocks which are used for sorting
;   204     0201  1 !
;   205     0202  1 ! Calling Sequence:
;   206     0203  1 !
;   207     0204  1 !       INIT_SORT()
;   208     0205  1 !
;   209     0206  1 ! Input Parameters:
;   210     0207  1 !       none
;   211     0208  1 !
;   212     0209  1 ! Implicit Inputs:
;   213     0210  1 !       none
;   214     0211  1 !
;   215     0212  1 ! Output Parameters:
;   216     0213  1 !       none
;   217     0214  1 !
;   218     0215  1 ! Implicit Outputs:
;   219     0216  1 !       none
;   220     0217  1 !
;   221     0218  1 ! Routines Called:
;   222     0219  1 !
;   223     0220  1 !       CONV$$GET_VM
;   224     0221  1 !
;   225     0222  1 ! Routine Value:
;   226     0223  1 !       none
;   227     0224  1 !
;   228     0225  1 ! Side Effects:
;   229     0226  1 !
;   230     0227  1 !       Clears the CONV$V_SORTINIT flag
;   231     0228  1 !
;   232     0229  1 !--
;   233     0230  1
;   234     0231  2     BEGIN
;   235     0232  2
;   236     0233  2     LOCAL
;   237     0234  2         BYTES,
;   238     0235  2         VM_POINTER;
;   239     0236  2
;   240     0237  2     ! If sort has already been initialized then exit
;   241     0238  2     !
;   242     0239  2     IF NOT .CONV$AB_FLAGS [ CONV$V_SORTINIT ]
;   243     0240  2     THEN
;   244     0241  3         BEGIN
;   245     0242  3
;   246     0243  3         CONV$AB_FLAGS [ CONV$V_SORTINIT ] = _SET;
;   247     0244  3
;   248     0245  3         ! Allocate name block buffers and the rfa buffer
;   249     0246  3         !
;   250     0247  3         BYTES = ESA_BUF_SIZ + RSA_BUF_SIZ + RFA_BUF_SIZ;
;   251     0248  3
;   252     0249  3         CONV$GL_RFA_BUFFER = CONV$$GET_VM( .BYTES );
;   253     0250  3         END
```

```
;  254          0251  3          VM_POINTER = .CONV$GL_RFA_BUFFER + RFA_BUF_SIZ;
;  255          0252  3
;  256          0253  3          ! Init the name block
;  257          0254  3          !
;  258        P 0255  3          $NAM_INIT ( NAM = RFA_NAM,
;  259        P 0256  3                      ESA = .VM_POINTER,
;  260        P 0257  3                      ESS = ESA_BUF_SIZ,
;  261        P 0258  3                      RSA = .VM_POINTER + ESA_BUF_SIZ,
;  262          0259  3                      RSS = RSA_BUF_SIZ );
;  263          0260  3
;  264          0261  3          ! Init the FAB
;  265          0262  3          !
;  266        P 0263  3          $FAB_INIT ( FAB = CONV$AB_RFA_FAB,
;  267        P 0264  3                      DNA = CONV_DEF_STR,
;  268        P 0265  3                      DNS = CONV_DEF_SIZ,
;  269        P 0266  3                      FAC = <BRO,GET>,
;  270        P 0267  3                      FNA = CONV_TMP_STR,
;  271        P 0268  3                      FNS = CONV_TMP_SIZ,
;  272        P 0269  3                      FOP = <CBT,SQO>,
;  273          0270  3                      NAM = RFA_NAM );
;  274          0271  3
;  275          0272  3          ! Init the RAB
;  276          0273  3          !
;  277        P 0274  3          $RAB_INIT ( RAB = CONV$AB_RFA_RAB,
;  278        P 0275  3                      FAB = CONV$AB_RFA_FAB,
;  279        P 0276  3                      ROP = BIO,
;  280        P 0277  3                      UBF = .CONV$GL_RFA_BUFFER,
;  281          0278  3                      USZ = RFA_BUF_SIZ );
;  282          0279  3
;  283          0280  2          END;
;  284          0281  2
;  285          0282  2          ! Set the record format and the record size
;  286          0283  2          !
;  287          0284  2          CONV$AB_RFA_FAB [ FAB$B_RFM ] = .RECORDFMT;
;  288          0285  2          CONV$AB_RFA_FAB [ FAB$W_MRS ] = .RECORDSIZ;
;  289          0286  2
;  290          0287  2          ! Clear the delete flag so that we don't delete the temp file this time
;  291          0288  2          !
;  292          0289  2          CONV$AB_RFA_FAB [ FAB$V_DLT ] = _CLEAR;
;  293          0290  2
;  294          0291  2          ! Signal create error
;  295          0292  2          !
;  296          0293  2          CONV$AB_RFA_FAB [ FAB$L_CTX ] = CONV$_CREA_ERR;
;  297          0294  2
;  298          0295  2          ! Create the file so that we get logical name direction to work and
;  299          0296  2          ! we pass a good file name to sort
;  300          0297  2          !
;  301          0298  2          $CREATE( FAB=CONV$AB_RFA_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;  302          0299  2
;  303          0300  2          $CLOSE( FAB=CONV$AB_RFA_FAB );
;  304          0301  2
;  305          0302  2          ! Set the delete flag so that we get rid of the temp file the next time
;  306          0303  2          ! we open it
;  307          0304  2          !
;  308          0305  2          CONV$AB_RFA_FAB [ FAB$V_DLT ] = _SET;
;  309          0306  2
;  310          0307  2          ! Stuff the expanded file name into the tempory file descriptor
```

```
;  311          0308  2     !
;  312          0309  2     CONV_TMP_DESC [ DSC$W_LENGTH ] = .RFA_NAM [ NAM$B_RSL ];
;  313          0310  2     CONV_TMP_DESC [ DSC$A_POINTER ] = .RFA_NAM [ NAM$C_RSA ];
;  314          0311  2
;  315          0312  2     RETURN
;  316          0313  2
;  317          0314  1     END;


                                                        .TITLE  CONV$SORT VAX-11 CONVERT
                                                        .IDENT  \V04-000\

                                                        .PSECT  _CONV$PLIT,NOWRT,NOEXE,  SHR,  PIC,2

                        4B  52  4F  57  56  4E  4F  43  00000 P.AAA:   .ASCII  \CONVWORK\
4D  54  2E  3A  48  43  54  41  52  43  53  24  53  59  53  00008 P.AAB:   .ASCII  \SYS$SCRATCH:.TMP\                ;
                                                    50  00017                                                       ;

                                                        .PSECT  _CONV$GLOBAL,NOEXE,  PIC,2

                                                00000 CONV$GL_RFA_BUFFER::
                                                                .BLKB   4
                                                00004 CONV$AB_RFA_FAB::
                                                                .BLKB   80
                                                00054 CONV$AB_RFA_RAB::
                                                                .BLKB   68

                                                        .PSECT  _CONV$OWN,NOEXE,  PIC,2

                                                00000 CONV_TMP_DESC:
                                                                .BLKB   8
                                                00008 TEMP_DESC:
                                                                .BLKB   8
                                                00010 RFA_NAM:  .BLKB   96
                                    12000020    00070 FOP:      .LONG   301989920                                   ;
                                                00074 FILETYPE:
                                                                .BLKB   1
                                                00075 RECORDFMT:
                                                                .BLKB   1
                                                00076 RECORDSIZ:
                                                                .BLKB   2

                                                        CONV_TMP_STR=       P.AAA
                                                        CONV_DEF_STR=       P.AAB
                                                        $RMS_PTR=           RFA_NAM
                                                        $RMS_PTR=           CONV$AB_RFA_FAB
                                                        $RMS_PTR=           CONV$AB_RFA_RAB
                                                                .EXTRN  CONVERT$_FACILITY
                                                                .EXTRN  CONV$_FAO_MAX, CONV$_BADBLK
                                                                .EXTRN  CONV$_BADLOGIC, CONV$_BADSORT
                                                                .EXTRN  CONV$_CONFQUAL, CONV$_CREATEDSTM
                                                                .EXTRN  CONV$_CREA_ERR, CONV$_DELPRI
                                                                .EXTRN  CONV$_DUP, CONV$_EXTN_ERR
                                                                .EXTRN  CONV$_FATALEXC, CONV$_FILLIM
                                                                .EXTRN  CONV$_IDX_LIM, CONV$_ILL_KEY
                                                                .EXTRN  CONV$_ILL_VALUE
                                                                .EXTRN  CONV$_INP_FILES
```

```
                                .EXTRN  CONV$_INSVIRMEM
                                .EXTRN  CONV$_INVBKT, CONV$_KEY
                                .EXTRN  CONV$_KEYREF, CONV$_LOADIDX
                                .EXTRN  CONV$_NARG, CONV$_NI
                                .EXTRN  CONV$_NOKEY, CONV$_NOTIDX
                                .EXTRN  CONV$_NOTSEQ, CONV$_NOWILD
                                .EXTRN  CONV$_ORDER, CONV$_OPENEXC
                                .EXTRN  CONV$_OPENIN, CONV$_OPENOUT
                                .EXTRN  CONV$_PAD, CONV$_PLV
                                .EXTRN  CONV$_PROERR, CONV$_PROL_WRT
                                .EXTRN  CONV$_READERR, CONV$_RSK
                                .EXTRN  CONV$_RSZ, CONV$_RTL
                                .EXTRN  CONV$_RTS, CONV$_SEQ
                                .EXTRN  CONV$_UDF_BKS, CONV$_UDF_BLK
                                .EXTRN  CONV$_VFC, CONV$_WRITEERR
                                .EXTRN  CONV$$GET_VM, CONV$$OPEN_IN
                                .EXTRN  CONV$$RMS_OPEN_ERROR
                                .EXTRN  CONV$$SET_KEY_DESC
                                .EXTRN  CONV$$SEARCH_FILE
                                .EXTRN  LIB$PUT_OUTPUT, SOR$BEGIN_SORT
                                .EXTRN  SOR$PASS_FILES, SOR$SORT_MERGE
                                .EXTRN  SOR$END_SORT, CONV$GL_SORT
                                .EXTRN  CONV$GL_WORK_F, CONV$AB_FLAGS
                                .EXTRN  CONV$AR_OUT_FILE_NAM
                                .EXTRN  CONV$GB_CURRENT_FILE
                                .EXTRN  CONV$GL_FILE_COUNT
                                .EXTRN  CONV$AR_PROLOGUE
                                .EXTRN  CONV$GW_MAX_REC_SIZ
                                .EXTRN  CONV$AB_IN_NAM, CONV$AB_IN_FAB
                                .EXTRN  CONV$AB_IN_RAB, CONV$AB_OUT_NAM
                                .EXTRN  CONV$AB_OUT_FAB
                                .EXTRN  CONV$AB_OUT_RAB
                                .EXTRN  SOR$M_STABLE, SOR$GK_RECORD
                                .EXTRN  SOR$GK_ADDRESS, SOR$GK_INDEX
                                .EXTRN  SYS$CREATE, SYS$CLOSE

                                .PSECT  _CONV$CODE,NOWRT,  SHR,  PIC,2

                        0FFC 00000 INIT_SORT:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 0195
                        5B      0000'   CF  9E  00002           MOVAB   $RMS_PTR, R8                    ;
                        57      0000'   CF  9E  00007           MOVAB   $RMS_PTR, R7                    ;
                03      0000G   CF          05  E1  0000C       BBC     #5, CONV$AB_FLAGS+2, 1$         ; 0239
                                        0098    31  00012       BRW     2$                              ;
                        0000G   CF          20  88  00015 1$:   BISB2   #32, CONV$AB_FLAGS+2            ; 0243
                        50      06A0        8F  3C  0001A       MOVZWL  #1696, BYTES                    ; 0247
                                            50  DD  0001F       PUSHL   BYTES                           ; 0249
                                        0000G   30  00021       BSBW    CONV$$GET_VM                    ;
                        5E                  04  C0  00024       ADDL2   #4, SP                          ;
                    FC  A7                  50  D0  00027       MOVL    R0, CONV$GL_RFA_BUFFER          ;
                56  FC  A7  00000600        8F  C1  0002B       ADDL3   #1536, CONV$GL_RFA_BUFFER, VM_POINTER   ; 0251
0060    8F      00          6E              00  2C  00034       MOVC5   #0, (SP), #0, #96, $RMS_PTR     ; 0259
                                            68  0003B           ;
                        68      6002        8F  B0  0003C       MOVW    #24578, $RMS_PTR                ;
                    02  A8          50      8F  90  00041       MOVB    #80, $RMS_PTR+2                 ;
                    04  A8          50      A6  9E  00046       MOVAB   80(R6), $RMS_PTR+4              ;
                    0A  A8          50      8F  90  0004B       MOVB    #80, $RMS_PTR+10                ;
```

```
                             0C   A8              56  D0 00050         MOVL    VM_POINTER, $RMS_PTR+12              ;
   0050  8F          00           6E              00  2C 00054         MOVC5   #0, (SP), #0, #80, $RMS_PTR           ; 0270
                                                  67     0005B
                                  67     5003     8F  B0 0005C         MOVW    #20483, $RMS_PTR                     ;
                             04   A7 00200040     8F  D0 00061         MOVL    #2097216, $RMS_PTR+4                 ;
                             16   A7       42     8F  90 00069         MOVB    #66, $RMS_PTR+22                     ;
                             1F   A7              02  90 0006E         MOVB    #2, $RMS_PTR+31                      ;
                             28   A7              68  9E 00072         MOVAB   RFA_NAM, $RMS_PTR+40                 ;
                             2C   A7     0000'    CF  9E 00076         MOVAB   CONV_TMP_STR, $RMS_PTR+44            ;
                             30   A7     0000'    CF  9E 0007C         MOVAB   CONV_DEF_STR, $RMS_PTR+48            ;
                             34   A7     1008     8F  B0 00082         MOVW    #4104, $RMS_PTR+52                   ;
   0044  8F          00           6E              00  2C 00088         MOVC5   #0, (SP), #0, #68, $RMS_PTR           ; 0278
                                           50     A7     0008F
                             50   A7     4401     8F  B0 00091         MOVW    #17409, $RMS_PTR                     ;
                             54   A7     0800     8F  3C 00097         MOVZWL  #2048, $RMS_PTR+4                    ;
                             70   A7     0600     8F  B0 0009D         MOVW    #1536, $RMS_PTR+32                   ;
                             74   A7       FC     A7  D0 000A3         MOVL    CONV$GL_RFA_BUFFER, $RMS_PTR+36      ;
                           008C   C7              67  9E 000A8         MOVAB   CONV$AB_RFA_FAB, $RMS_PTR+60         ;
                             1F   A7       65     A8  90 000AD  2$:    MOVB    RECORDFMT, CONV$AB_RFA_FAB+31        ; 0284
                             36   A7       66     A8  B0 000B2         MOVW    RECORDSIZ, CONV$AB_RFA_FAB+54        ; 0285
                             05   A7       80     8F  8A 000B7         BICB2   #128, CONV$AB_RFA_FAB+5              ; 0289
                             18   A7 00000000G    8F  D0 000BC         MOVL    #CONV$_CREA_ERR, CONV$AB_RFA_FAB+24  ; 0293
                                         0000G    CF  9F 000C4         PUSHAB  CONV$$RMS_OPEN_ERROR                 ; 0298
                                                  57  DD 000C8         PUSHL   R7                                   ;
                      00000000G   00              02  FB 000CA         CALLS   #2, SYS$CREATE                       ;
                                                  57  DD 000D1         PUSHL   R7                                   ; 0300
                      00000000G   00              01  FB 000D3         CALLS   #1, SYS$CLOSE                        ;
                             05   A7       80     8F  88 000DA         BISB2   #128, CONV$AB_RFA_FAB+5              ; 0305
                             F0   A8       03     A8  9B 000DF         MOVZBW  RFA_NAM+3, CONV_TMP_DESC             ; 0309
                             F4   A8       04     A8  D0 000E4         MOVL    RFA_NAM+4, CONV_TMP_DESC+4           ; 0310
                                                      04 000E9         RET                                          ; 0314

; Routine Size:  234 bytes,    Routine Base:  _CONV$CODE + 0000
```

```
;  319     0315  1 %SBTTL  'SORT_PRIMARY'
;  320     0316  1 GLOBAL ROUTINE  CONV$$SORT_PRIMARY : CL$SORT_PRIMARY =
;  321     0317  1 !++
;  322     0318  1 !
;  323     0319  1 ! Functional Description:
;  324     0320  1 !
;  325     0321  1 !       This routine will sort the input file, pointed to by in_fab, according
;  326     0322  1 !       to the primary key of the output file.
;  327     0323  1 !
;  328     0324  1 ! Calling Sequence:
;  329     0325  1 !
;  330     0326  1 !       CONV$$SORT_PRIMARY()
;  331     0327  1 !
;  332     0328  1 ! Input Parameters:
;  333     0329  1 !       none
;  334     0330  1 !
;  335     0331  1 ! Implicit Inputs:
;  336     0332  1 !
;  337     0333  1 !       input and output rms blocks
;  338     0334  1 !
;  339     0335  1 ! Output Parameters:
;  340     0336  1 !       none
;  341     0337  1 !
;  342     0338  1 ! Implicit Outputs:
;  343     0339  1 !       none
;  344     0340  1 !
;  345     0341  1 ! Routines Called:
;  346     0342  1 !
;  347     0343  1 !       INIT_SORT
;  348     0344  1 !       SOR$PASS_FILES
;  349     0345  1 !       SORT_ERROR
;  350     0346  1 !       CONV$$SEARCH_FILE
;  351     0347  1 !       SET_UP_SORT
;  352     0348  1 !       SOR$SORT_MERGE
;  353     0349  1 !       SOR$END_SORT
;  354     0350  1 !
;  355     0351  1 ! Routine Value:
;  356     0352  1 !
;  357     0353  1 !       Success of random errors
;  358     0354  1 !
;  359     0355  1 ! Side Effects:
;  360     0356  1 !
;  361     0357  1 !       Open the rfa file if CONV$V_RFA is set
;  362     0358  1 !
;  363     0359  1 !--
;  364     0360  1
;  365     0361  2     BEGIN
;  366     0362  2
;  367     0363  2     DEFINE_KEY_DESC;
;  368     0364  2
;  369     0365  2     LOCAL
;  370     0366  2         IN_DEVICE       : BLOCK [ 1,LONG ],
;  371     0367  2         RFA             : LONG;
;  372     0368  2
;  373     0369  2     ! Set the key descriptor to key = 0 (always in prologue)
;  374     0370  2     !
;  375     0371  2     KEY_DESC = .CONV$AR_PROLOGUE;
```

```
;   376     0372  2
;   377     0373  2        ! If the input file is open close it
;   378     0374  2        !
;   379     0375  2        IF .CONV$AB_FLAGS [ CONV$V_IN ]
;   380     0376  2        THEN
;   381     0377  2            BEGIN
;   382     0378  2            $DISCONNECT( RAB=CONV$AB_IN_RAB );
;   383     0379  2            $CLOSE( FAB=CONV$AB_IN_FAB );
;   384     0380  2            CONV$AB_FLAGS [ CONV$V_IN ] = _CLEAR
;   385     0381  2            END;
;   386     0382  2
;   387     0383  2        IN_DEVICE = .CONV$AB_IN_FAB [ FAB$L_DEV ];
;   388     0384  2
;   389     0385  2        ! If the device char. are zero (process perminant files) or
;   390     0386  2        ! if the input file is not from disk or
;   391     0387  2        ! it is a record orented device (terminals) or
;   392     0388  2        ! it's a network file or
;   393     0389  2        ! it's a terminal (be reduntandt) or
;   394     0390  2        ! there is more than one input file
;   395     0391  2        ! then we do a normal record sort otherwise we do a RFA sort to save time
;   396     0392  2        !
;   397     0393  2        IF ( .IN_DEVICE EQLU 0 ) OR
;   398     0394  2            .IN_DEVICE [ DEV$V_SQD ] OR
;   399     0395  2            .IN_DEVICE [ DEV$V_NET ] OR
;   400     0396  2            .IN_DEVICE [ DEV$V_REC ] OR
;   401     0397  2            .IN_DEVICE [ DEV$V_TRM ] OR
;   402     0398  2            ( .CONV$GL_FILE_COUNT GTR 1 )
;   403     0399  2        THEN
;   404     0400  2            BEGIN
;   405     0401  2            RFA = _CLEAR;
;   406     0402  2            RECORDFMT = FAB$C_VAR;
;   407     0403  2            RECORDSIZ = 0;
;   408     0404  2            END
;   409     0405  2        ELSE
;   410     0406  2            BEGIN
;   411     0407  2            RFA = _SET;
;   412     0408  2            RECORDFMT = FAB$C_FIX;
;   413     0409  2            RECORDSIZ = 6;
;   414     0410  2            END;
;   415     0411  2
;   416     0412  2        ! Initialize the RMS blocks used in the sort
;   417     0413  2        !
;   418     0414  2        INIT_SORT();
;   419     0415  2
;   420     0416  2        ! Pass the file names 1st input and output
;   421     0417  2        !
;   422     0418  2        TEMP_DESC [ DSC$W_LENGTH ] = .CONV$AB_IN_FAB [ FAB$B_FNS ];
;   423     0419  2        TEMP_DESC [ DSC$A_POINTER ] = .CONV$AB_IN_FAB [ FAB$L_FNA ];
;   424     0420  2
;   425     0421  2        SOR$PASS_FILES( TEMP_DESC,
;   426     0422  2                        CONV_TMP_DESC,
;   427     0423  2                        FILETYPE,
;   428     0424  2                        RECORDFMT,
;   429     0425  2                        0,
;   430     0426  2                        0,
;   431     0427  2                        0,
;   432     0428  2                        0,
```

```
2                       FOP );
2
2       CONV$GB_CURRENT_FILE = 1;
2
2       ! Pass the rest of the input names
2       !
3       UNTIL .CONV$GB_CURRENT_FILE GTR ( .CONV$GL_FILE_COUNT - 1 )
3       DO
3           BEGIN
3
3           ! Parse and search for the file (This uses the IN_FAB and IN_NAM
3           ! since they are not used again)
3           !
3           RET_ON_ERROR( CONV$$SEARCH_FILE() );
3
3           ! Pass the file spec
3           !
3           TEMP_DESC [ DSC$W_LENGTH ] = .CONV$AB_IN_FAB [ FAB$B_FNS ];
3           TEMP_DESC [ DSC$A_POINTER ] = .CONV$AB_IN_FAB [ FAB$L_FNA ];
3
3           SOR$PASS_FILES( TEMP_DESC );
3
3           CONV$GB_CURRENT_FILE = .CONV$GB_CURRENT_FILE + 1
3
3           END;
2
2       ! If useing rfa file as index file do an index sort else do record sort
2       !
2       IF .RFA
2       THEN
2           SET_UP_SORT( SOR$GK_ADDRESS )
2       ELSE
2           SET_UP_SORT( SOR$GK_RECORD );
2
2       ! Do the sort
2       !
2       SOR$SORT_MERGE();
2
2       SOR$END_SORT();
2
2       ! Reopen the correct input files
2       !
2       IF .RFA
2       THEN
3           BEGIN
3
3           ! OPEN the input file and the new RFA file
3           !
3           RET_ON_ERROR( CONV$$OPEN_IN() );
3
3           ! Connect the additional file containing the RFAs pointing th the real
3           ! file
3           !
3           $OPEN( FAB=CONV$AB_RFA_FAB );
3           $CONNECT( RAB=CONV$AB_RFA_RAB );
3
3           CONV$AB_FLAGS [ CONV$V_RFA ] = _SET;
```

```
3
3              ! Set access to the real input file to RFA
3              !
3              CONV$AB_IN_RAB [ RAB$B_RAC ] = RAB$C_RFA
3
3              END
2          ELSE
2
2              ! OPEN the sorted file as if it was the input file
2              !
3              BEGIN
3
3              ! The real input RAB points to the RFA FAB
3              !
3              CONV$AB_IN_RAB [ RAB$L_FAB ] = CONV$AB_RFA_FAB;
3
3              ! Open the RFA fab which is the new sorted input file NOTE: This is
3              ! not a file of RFAs an above
3              !
3              $OPEN( FAB=CONV$AB_RFA_FAB );
3              $CONNECT( RAB=CONV$AB_IN_RAB );
3
3              CONV$AB_FLAGS [ CONV$V_SOR ] = _SET
3
3              END;
2
2          ! Since it only makes sence to sort once
2          !
2          CONV$GL_SORT = _CLEAR;
2
2          RETURN CONV$_SUCCESS
2
1          END;
```

```
                                                  .EXTRN  SYS$DISCONNECT, SYS$OPEN
                                                  .EXTRN  SYS$CONNECT

                            52  DD 00000 CONV$$SORT_PRIMARY::
                                                  PUSHL   R2                                  ; 0316
              5B     0000G  CF  D0 00002          MOVL    CONV$AR_PROLOGUE, KEY_DESC          ; 0371
              1B     0000G  CF  E9 00007          BLBC    CONV$AB_FLAGS+2, 1$                 ; 0375
                     0000G  CF  9F 0000C          PUSHAB  CONV$AB_IN_RAB                      ; 0378
  00000000G   00            01  FB 00010          CALLS   #1, SYS$DISCONNECT
                     0000G  CF  9F 00017          PUSHAB  CONV$AB_IN_FAB                      ; 0379
  00000000G   00            01  FB 0001B          CALLS   #1, SYS$CLOSE
      0000G   CF            01  8A 00022          BICB2   #1, CONV$AB_FLAGS+2                 ; 0380
              50     0000G  CF  D0 00027 1$:      MOVL    CONV$AB_IN_FAB+64, IN_DEVICE        ; 0383
                            16  13 0002C          BEQL    2$                                  ; 0393
       12     50            05  E0 0002E          BBS     #5, IN_DEVICE, 2$                   ; 0394
       0E     50            0D  E0 00032          BBS     #13, IN_DEVICE, 2$                  ; 0395
              08            50  E8 00036          BLBS    IN_DEVICE, 2$                       ; 0396
       07     50            02  E0 00039          BBS     #2, IN_DEVICE, 2$                   ; 0397
              01     0000G  CF  D1 0003D          CMPL    CONV$GL_FILE_COUNT, #1              ; 0398
                            0D  15 00042          BLEQ    3$
                            52  D4 00044 2$:      CLRL    RFA                                 ; 0401
```

```
                                0000'  CF                 02  90 00046          MOVB    #2, RECORDFMT                           ;0402
                                                   0000'  CF  B4 0004B          CLRW    RECORDSIZ                               ;0403
                                                          0D  11 0004F          BRB     4$                                      ;0393
                                    52                    01  D0 00051  3$:     MOVL    #1, RFA                                 ;0407
                                0000'  CF                 01  90 00054          MOVB    #1, RECORDFMT                           ;0408
                                0000'  CF                 06  B0 00059          MOVW    #6, RECORDSIZ                           ;0409
                                FEB3   CF                 00  FB 0005E  4$:     CALLS   #0, INIT_SORT                           ;0414
                                0000'  CF          0000G  CF  9B 00063          MOVZBW  CONV$AB_IN_FAB+52, TEMP_DESC            ;0418
                                0000'  CF          0000G  CF  D0 0006A          MOVL    CONV$AB_IN_FAB+44, TEMP_DESC+4          ;0419
                                                   0000'  CF  9F 00071          PUSHAB  FOP                                     ;0421
                                                          7E  7C 00075          CLRQ    -(SP)                                   ;
                                                          7E  7C 00077          CLRQ    -(SP)                                   ;
                                                   0000'  CF  9F 00079          PUSHAB  RECORDFMT                               ;
                                                   0000'  CF  9F 0007D          PUSHAB  FILETYPE                                ;
                                                   0000'  CF  9F 00081          PUSHAB  CONV_TMP_DESC                           ;
                                                   0000'  CF  9F 00085          PUSHAB  TEMP_DESC                               ;
                            00000000G  00                    09  FB 00089          CALLS   #9, SOR$PASS_FILES                      ;
                                 0000G  CF                   01  90 00090          MOVB    #1, CONV$GB_CURRENT_FILE                ;0431
                           50    0000G  CF                   01  C3 00095  5$:     SUBL3   #1, CONV$GL_FILE_COUNT, R0              ;0435
                     50         0000G  CF        08         00  ED 0009B          CMPZV   #0, #8, CONV$GB_CURRENT_FILE, R0        ;
                                                          27  14 000A2          BGTR    6$                                      ;
                                 0000G  CF                   00  FB 000A4          CALLS   #0, CONV$SEARCH_FILE                    ;0442
                                        4C                   50  E9 000A9          BLBC    STATUS, 9$                              ;
                                0000'  CF          0000G  CF  9B 000AC          MOVZBW  CONV$AB_IN_FAB+52, TEMP_DESC            ;0446
                                0000'  CF          0000G  CF  D0 000B3          MOVL    CONV$AB_IN_FAB+44, TEMP_DESC+4          ;0447
                                                   0000'  CF  9F 000BA          PUSHAB  TEMP_DESC                               ;0449
                            00000000G  00                    01  FB 000BE          CALLS   #1, SOR$PASS_FILES                      ;
                                                   0000G  CF  96 000C5          INCB    CONV$GB_CURRENT_FILE                    ;0451
                                                          CA  11 000C9          BRB     5$                                      ;
                                    08                    52  E9 000CB  6$:     BLBC    RFA, 7$                                 ;0457
                                        00000000G  8F        DD 000CE          PUSHL   #SOR$GK_ADDRESS                         ;0459
                                                          06  11 000D4          BRB     8$                                      ;
                                        00000000G  8F        DD 000D6  7$:     PUSHL   #SOR$GK_RECORD                          ;0461
                                              0000V 30        000DC  8$:     BSBW    SET_UP_SORT                             ;
                                    5E                    04  C0 000DF          ADDL2   #4, SP                                  ;
                            00000000G  00                    00  FB 000E2          CALLS   #0, SOR$SORT_MERGE                      ;0465
                            00000000G  00                    00  FB 000E9          CALLS   #0, SOR$END_SORT                        ;0467
                                    2A                    52  E9 000F0          BLBC    RFA, 10$                                ;0471
                                 0000G  CF                   00  FB 000F3          CALLS   #0, CONV$$OPEN_IN                       ;0477
                                        4B                   50  E9 000F8  9$:     BLBC    STATUS, 12$                             ;
                                                   0000'  CF  9F 000FB          PUSHAB  CONV$AB_RFA_FAB                         ;0482
                            00000000G  00                    01  FB 000FF          CALLS   #1, SYS$OPEN                            ;
                                                   0000'  CF  9F 00106          PUSHAB  CONV$AB_RFA_RAB                         ;0483
                            00000000G  00                    01  FB 0010A          CALLS   #1, SYS$CONNECT                         ;
                                 0000G  CF                   10  88 00111          BISB2   #16, CONV$AB_FLAGS+2                    ;0485
                                 0000G  CF                   02  90 00116          MOVB    #2, CONV$AB_IN_RAB+30                   ;0489
                                                          22  11 0011B          BRB     11$                                     ;
                                 0000G  CF         0000'  CF  9E 0011D  10$:     MOVAB   CONV$AB_RFA_FAB, CONV$AB_IN_RAB+60      ;0500
                                                   0000'  CF  9F 00124          PUSHAB  CONV$AB_RFA_FAB                         ;0505
                            00000000G  00                    01  FB 00128          CALLS   #1, SYS$OPEN                            ;
                                                   0000G  CF  9F 0012F          PUSHAB  CONV$AB_IN_RAB                          ;0506
                            00000000G  00                    01  FB 00133          CALLS   #1, SYS$CONNECT                         ;
                                 0000G  CF                   08  88 0013A          BISB2   #8, CONV$AB_FLAGS+2                     ;0508
                                                   0000G  CF  D4 0013F  11$:     CLRL    CONV$GL_SORT                            ;0514
                                    50                    01  D0 00143          MOVL    #1, R0                                  ;0516
                                                          04  BA 00146  12$:     POPR    #^M<R2>                                 ;0518
                                                          05  00148          RSB                                             ;
```

; Routine Size: 329 bytes,   Routine Base: _CONV$CODE + 00EA

```
;   524          0519  1 %SBTTL  'SORT_SECONDARY'
;   525          0520  1 GLOBAL ROUTINE  CONV$$SORT_SECONDARY : CL$SORT_SECONDARY =
;   526          0521  1 !++
;   527          0522  1 !
;   528          0523  1 ! Functional Description:
;   529          0524  1 !
;   530          0525  1 !       This routine will sort the OUTPUT file according to a specified
;   531          0526  1 !       key of the OUTPUT file.
;   532          0527  1 !
;   533          0528  1 ! Calling Sequence:
;   534          0529  1 !
;   535          0530  1 !       CONV$$SORT_SECONDARY()
;   536          0531  1 !
;   537          0532  1 ! Input Parameters:
;   538          0533  1 !       none
;   539          0534  1 !
;   540          0535  1 ! Implicit Inputs:
;   541          0536  1 !       none
;   542          0537  1 !
;   543          0538  1 ! Output Parameters:
;   544          0539  1 !       none
;   545          0540  1 !
;   546          0541  1 ! Implicit Outputs:
;   547          0542  1 !       none
;   548          0543  1 !
;   549          0544  1 ! Routines Called:
;   550          0545  1 !
;   551          0546  1 !       INIT_SORT
;   552          0547  1 !       SOR$PASS_FILES
;   553          0548  1 !       SET_UP_SORT
;   554          0549  1 !       SOR$SORT_MERGE
;   555          0550  1 !       SOR$END_SORT
;   556          0551  1 !
;   557          0552  1 ! Routine Value:
;   558          0553  1 !
;   559          0554  1 !       Success or random errors
;   560          0555  1 !
;   561          0556  1 ! Side Effects:
;   562          0557  1 !
;   563          0558  1 !       Closes and reopens the output file
;   564          0559  1 !       Closes the rfa file if it was open then opens it
;   565          0560  1 !
;   566          0561  1 !--
;   567          0562  1
;   568          0563  2     BEGIN
;   569          0564  2
;   570          0565  2     DEFINE_KEY_DESC;
;   571          0566  2
;   572          0567  2     ! If the RFA file was open close it. The file will be used as output of sort.
;   573          0568  2     !
;   574          0569  2     IF .CONV$AB_FLAGS [ CONV$V_RFA ]
;   575          0570  2     THEN
;   576          0571  3         BEGIN
;   577          0572  3         ERRCHK( $DISCONNECT( RAB=CONV$AB_RFA_RAB ),CONV$_BADLOGIC );
;   578          0573  3         ERRCHK( $CLOSE( FAB=CONV$AB_RFA_FAB ),CONV$_BADLOGIC );
;   579          0574  3
;   580          0575  3         CONV$AB_FLAGS [ CONV$V_RFA ] = _CLEAR;
```

```
;   581     0576  3
;   582     0577  3          ! Also remove the entry in the directory
;   583     0578  3          !
;   584     0579  4          $ERASE( FAB=CONV$AB_RFA_FAB )
;   585     0580  4
;   586     0581  2          END;
;   587     0582  2
;   588     0583  2      ! Secondary key sorts are always tag sorts therefore we need a var. file
;   589     0584  2      !
;   590     0585  2      RECORDFMT = FAB$C_VAR;
;   591     0586  2      RECORDSIZ = 0;
;   592     0587  2
;   593     0588  2      ! Init sort if necc. and get a file name
;   594     0589  2      !
;   595     0590  2      INIT_SORT();
;   596     0591  2
;   597     0592  2      ! To conserve space ect. use the RFA fab and rab therefore reset
;   598     0593  2      ! the RFA rab so we can do record I/O on it.  We can use the rfa buffer
;   599     0594  2      ! since it is at least 512 bytes long and a key is only 256 + 6 byte rfa
;   600     0595  2      !
;   601     0596  2      ! Clear the BIO flag
;   602     0597  2      !
;   603     0598  2      CONV$AB_RFA_RAB [ RAB$V_BIO ] = _CLEAR;
;   604     0599  2
;   605     0600  2      ! Close the current output file so that SORT can get at it
;   606     0601  2      !
;   607     0602  2      $DISCONNECT( RAB=CONV$AB_OUT_RAB );
;   608     0603  2      ERRCHK( $CLOSE( FAB=CONV$AB_OUT_FAB ),CONV$_BADLOGIC );
;   609     0604  2
;   610     0605  2      CONV$AB_FLAGS [ CONV$V_OUT ] = _CLEAR;
;   611     0606  2
;   612     0607  2      ! Pass the file names
;   613     0608  2      !
;   614     0609  2      ! To avoid some file name problems pass the expanded string of the
;   615     0610  2      ! output file
;   616     0611  2      !
;   617     0612  2      TEMP_DESC [ DSC$W_LENGTH ] = .CONV$AB_OUT_NAM [ NAM$B_RSL ];
;   618     0613  2      TEMP_DESC [ DSC$A_POINTER ] = .CONV$AB_OUT_NAM [ NAM$[_RSA ];
;   619     0614  2
;   620     0615  2      SOR$PASS_FILES( TEMP_DESC,
;   621     0616  2                      CONV_TMP_DESC,
;   622     0617  2                      FILETYPE,
;   623     0618  2                      RECORDFMT,
;   624     0619  2                      0,
;   625     0620  2                      0,
;   626     0621  2                      0,
;   627     0622  2                      0,
;   628     0623  2                      FOP );
;   629     0624  2
;   630     0625  2      ! Get ready to do a index sort of the file
;   631     0626  2      !
;   632     0627  2      SET_UP_SORT( SOR$GK_INDEX );
;   633     0628  2
;   634     0629  2      ! Start the sort and finish it
;   635     0630  2      !
;   636     0631  2      SOR$SORT_MERGE();
;   637     0632  2      SOR$END_SORT();
```

```
;  638       0633 2
;  639       0634 2        ! ReOPEN the output file and the new RFA-INDEX file
;  640       0635 2        !
;  641       0636 2        $OPEN( FAB=CONV$AB_OUT_FAB );
;  642       0637 2        $CONNECT( RAB=CONV$AB_OUT_RAB );
;  643       0638 2        CONV$AB_FLAGS [ CONV$V_OUT ] = _SET;
;  644       0639 2
;  645       0640 2        $OPEN( FAB=CONV$AB_RFA_FAB );
;  646       0641 2        $CONNECT( RAB=CONV$AB_RFA_RAB );
;  647       0642 2        CONV$AB_FLAGS [ CONV$V_RFA ] = _SET;
;  648       0643 2
;  649       0644 2        RETURN SS$_NORMAL
;  650       0645 1        END;


                                                         .EXTRN  SYS$ERASE

                                        52 DD 00000 CONV$$SORT_SECONDARY::
                                                         PUSHL   R2                                     ; 0520
           32     0000G  CF             04 E1 00002      BBC     #4, CONV$AB_FLAGS+2, 1$                ; 0569
                                  0000'  CF 9F 00008      PUSHAB  CONV$AB_RFA_RAB                        ; 0572
              00000000G  00             01 FB 0000C      CALLS   #1, SYS$DISCONNECT                     ;
                         52             50 D0 00013      MOVL    R0, STATUS                             ;
                         50             52 E9 00016      BLBC    STATUS, 2$                             ;
                                  0000'  CF 9F 00019      PUSHAB  CONV$AB_RFA_FAB                        ; 0573
              00000000G  00             01 FB 0001D      CALLS   #1, SYS$CLOSE                          ;
                         52             50 D0 00024      MOVL    R0, STATUS                             ;
                         3F             52 E9 00027      BLBC    STATUS, 2$                             ;
                  0000G  CF             10 8A 0002A      BICB2   #16, CONV$AB_FLAGS+2                   ; 0575
                                  0000'  CF 9F 0002F      PUSHAB  CONV$AB_RFA_FAB                        ; 0579
              00000000G  00             01 FB 00033      CALLS   #1, SYS$ERASE                          ;
                  0000'  CF             02 90 0003A 1$:  MOVB    #2, RECORDFMT                          ; 0585
                                  0000'  CF B4 0003F      CLRW    RECORDSIZ                              ; 0586
                   FD85  CF             00 FB 00043      CALLS   #0, INIT_SORT                          ; 0590
                  0000'  CF             08 8A 00048      BICB2   #8, CONV$AB_RFA_RAB+5                  ; 0598
                                  0000G  CF 9F 0004D      PUSHAB  CONV$AB_OUT_RAB                        ; 0602
              00000000G  00             01 FB 00051      CALLS   #1, SYS$DISCONNECT                     ;
                                  0000G  CF 9F 00058      PUSHAB  CONV$AB_OUT_FAB                        ; 0603
              00000000G  00             01 FB 0005C      CALLS   #1, SYS$CLOSE                          ;
                         52             50 D0 00063      MOVL    R0, STATUS                             ;
                         13             52 E8 00066      BLBS    STATUS, 3$                             ;
                             00000000G  8F DD 00069 2$:  PUSHL   #CONV$_BADLOGIC                        ;
              00000000G  00             01 FB 0006F      CALLS   #1, LIB$SIGNAL                         ;
                         50             52 D0 00076      MOVL    STATUS, R0                             ;
                                     0085 31 00079      BRW     4$                                     ;
                  0000G  CF             02 8A 0007C 3$:  BICB2   #2, CONV$AB_FLAGS+2                    ; 0605
                  0000'  CF      0000G  CF 9B 00081      MOVZBW  CONV$AB_OUT_NAM+3, TEMP_DESC           ; 0612
                  0000'  CF      0000G  CF D0 00088      MOVL    CONV$AB_OUT_NAM+4, TEMP_DESC+4         ; 0613
                                  0000'  CF 9F 0008F      PUSHAB  FOP                                    ; 0615
                                        7E 7C 00093      CLRQ    -(SP)                                  ;
                                        7E 7C 00095      CLRQ    -(SP)                                  ;
                                  0000'  CF 9F 00097      PUSHAB  RECORDFMT                              ;
                                  0000'  CF 9F 0009B      PUSHAB  FILETYPE                               ;
                                  0000'  CF 9F 0009F      PUSHAB  CONV_TMP_DESC                          ;
                                  0000'  CF 9F 000A3      PUSHAB  TEMP_DESC                              ;
              00000000G  00             09 FB 000A7      CALLS   #9, SOR$PASS_FILES                     ;
```

```
                        00000000G  8F  DD 000AE          PUSHL   #SOR$GK_INDEX                           ; 0627
                             0000V 30  000B4          BSBW    SET_UP_SORT                             ;
                        5E         04  C0  000B7          ADDL2   #4, SP                              ;
                00000000G  00      00  FB  000BA          CALLS   #0, SOR$SORT_MERGE                  ; 0631
                00000000G  00      00  FB  000C1          CALLS   #0, SOR$END_SORT                    ; 0632
                           0000G   CF  9F  000C8          PUSHAB  CONV$AB_OUT_FAB                     ; 0636
                00000000G  00      01  FB  000CC          CALLS   #1, SYS$OPEN                        ;
                           0000G   CF  9F  000D3          PUSHAB  CONV$AB_OUT_RAB                     ; 0637
                00000000G  00      01  FB  000D7          CALLS   #1, SYS$CONNECT                     ;
                     0000G  CF     02  88  000DE          BISB2   #2, CONV$AB_FLAGS+2                 ; 0638
                           0000'   CF  9F  000E3          PUSHAB  CONV$AB_RFA_FAB                     ; 0640
                00000000G  00      01  FB  000E7          CALLS   #1, SYS$OPEN                        ;
                           0000'   CF  9F  000EE          PUSHAB  CONV$AB_RFA_RAB                     ; 0641
                00000000G  00      01  FB  000F2          CALLS   #1, SYS$CONNECT                     ;
                     0000G  CF     10  88  000F9          BISB2   #16, CONV$AB_FLAGS+2                ; 0642
                        50         01  D0  000FE          MOVL    #1, R0                              ; 0644
                                   04  BA  00101 4$:       POPR    #^M<R2>                             ; 0645
                                   05  00103              RSB                                         ;

; Routine Size:  260 bytes,    Routine Base:  _CONV$CODE + 0233
```

```
;  652          0646  1 %SBTTL  'SET_UP_SORT'
;  653          0647  1 ROUTINE SET_UP_SORT ( S_TYPE ) : CL$JSB_REG_11 NOVALUE =
;  654          0648  1 !++
;  655          0649  1 !
;  656          0650  1 ! Functional Description:
;  657          0651  1 !
;  658          0652  1 !       Initializes the control blocks for the sort utility
;  659          0653  1 !
;  660          0654  1 ! Calling Sequence:
;  661          0655  1 !
;  662          0656  1 !       SET_UP_SORT( sort_type )
;  663          0657  1 !
;  664          0658  1 ! Input Parameters:
;  665          0659  1 !
;  666          0660  1 !       sort_type - The sort code for the type of sort wanted.  Valid
;  667          0661  1 !                   codes are:
;  668          0662  1 !                       SOR$GK_RECORD = Record sort (Primary key from non-
;  669          0663  1 !                                       disk device or multiple input files)
;  670          0664  1 !                       SOR$GK_ADDRESS = Rfa sort (Primary key form disk)
;  671          0665  1 !                       SOR$GK_INDEX = Index sort (Secondary keys only)
;  672          0666  1 !
;  673          0667  1 ! Implicit Inputs:
;  674          0668  1 !
;  675          0669  1 !       KEY_DESC
;  676          0670  1 !
;  677          0671  1 ! Output Parameters:
;  678          0672  1 !       none
;  679          0673  1 !
;  680          0674  1 ! Implicit Outputs:
;  681          0675  1 !       none
;  682          0676  1 !
;  683          0677  1 ! Routines Called:
;  684          0678  1 !
;  685          0679  1 !       SOR$BEGIN_SORT
;  686          0680  1 !
;  687          0681  1 ! Routine Value:
;  688          0682  1 !
;  689          0683  1 !       Success of error from sor$begin_sort
;  690          0684  1 !
;  691          0685  1 ! Side Effects:
;  692          0686  1 !       none
;  693          0687  1 !
;  694          0688  1 !--
;  695          0689  1
;  696          0690  2     BEGIN
;  697          0691  2
;  698          0692  2     DEFINE_KEY_DESC;
;  699          0693  2
;  700          0694  2     ! Sort parameters
;  701          0695  2     !
;  702          0696  2     OWN
;  703          0697  2         KEY_BUFFER      : VECTOR [ 33,WORD ],
;  704          0698  2         LRL             : WORD,
;  705          0699  2         SORT_OPTIONS    : LONG,
;  706          0700  2         SORT_TYPE       : BYTE,
;  707          0701  2         WORK_FILES      : BYTE;
;  708          0702  2
```

```
;  709          0703  2      BIND
;  710          0704  2          SEGMENTS        = KEY_BUFFER [ 0 ] : WORD,
;  711          0705  2          SORT_KEY        = KEY_BUFFER [ 1 ] : BLOCKVECTOR [ 8,4,WORD ];
;  712          0706  2
;  713          0707  2      LOCAL
;  714          0708  2          KEY_TYPE;
;  715          0709  2
;  716          0710  2      SORT_TYPE    = .S_TYPE;
;  717          0711  2      WORK_FILES   = .CONV$GL_WORK_F;
;  718          0712  2      LRL          = .CONV$GW_MAX_REC_SIZ;
;  719          0713  2
;  720          0714  2      ! If the key allows dups do a stable sort
;  721          0715  2      !
;  722          0716  2      IF .KEY_DESC [ KEY$V_DUPKEYS ]
;  723          0717  2      THEN
;  724          0718  2          SORT_OPTIONS = SOR$M_STABLE
;  725          0719  2      ELSE
;  726          0720  2          SORT_OPTIONS = _CLEAR;
;  727          0721  2
;  728          0722  2      ! Get the number of segments
;  729          0723  2      !
;  730          0724  2      SEGMENTS = .KEY_DESC [ KEY$B_SEGMENTS ];
;  731          0725  2
;  732          0726  2      ! Find the key type from the key descriptor and set key_type to the
;  733          0727  2      ! appropiate SORT-32 code
;  734          0728  2      !
;  735          0729  3      KEY_TYPE = ( SELECTONE .KEY_DESC [ KEY$B_DATATYPE ] OF
;  736          0730  3          SET
;  737          0731  3          [ KEY$C_STRING ]        : DSC$K_DTYPE_T;
;  738          0732  3          [ KEY$C_SGNWORD ]       : DSC$K_DTYPE_W;
;  739          0733  3          [ KEY$C_SGNLONG ]       : DSC$K_DTYPE_L;
;  740          0734  3          [ KEY$C_SGNQUAD ]       : DSC$K_DTYPE_Q;
;  741          0735  3          [ KEY$C_UNSGNWORD ]     : DSC$K_DTYPE_WU;
;  742          0736  3          [ KEY$C_UNSGNLONG ]     : DSC$K_DTYPE_LU;
;  743          0737  3          [ KEY$C_UNSGNQUAD ]     : DSC$K_DTYPE_QU;
;  744          0738  3          [ KEY$C_PACKED ]        : DSC$K_DTYPE_P;
;  745          0739  3          TES );
;  746          0740  2
;  747          0741  2      ! Load the sort parameter block with the right stuff for each segment
;  748          0742  2      !
;  749          0743  2      INCR I FROM 0 TO ( .SEGMENTS - 1 ) BY 1
;  750          0744  2      DO
;  751          0745  2          BEGIN
;  752          0746  2          SORT_KEY [ .I,SORTKEY$W_TYPE ] = .KEY_TYPE;
;  753          0747  2          SORT_KEY [ .I,SORTKEY$W_ORDER ] = 0;
;  754          0748  2
;  755          0749  2          ! NOTE: The 28 is the offset to the first segment position descriptor
;  756          0750  2          !       in the key descriptor block the 44 is the offset to the segment
;  757          0751  2          !       size.  If the macros for these ever change, ie. KEY$W_POSITION and
;  758          0752  2          !       KEY$B_SIZE, the code offsets here must be changed!
;  759          0753  2          !
;  760          0754  2          SORT_KEY [ .I,SORTKEY$W_START ] = .KEY_DESC [ ( 28 + (.I*2) ),WORD_U ];
;  761          0755  2          SORT_KEY [ .I,SORTKEY$W_LENGH ] = .KEY_DESC [ ( 44 + .I   ),BYTE_U ];
;  762          0756  2
;  763          0757  2          ! If the key is packed decimal then sort wants the size in nibbles NOT
;  764          0758  2          ! counting the sign
;  765          0759  3          !
```

```
;  766     0760  3          IF .KEY_DESC [ KEY$B_DATATYPE ] EQLU KEY$C_PACKED
;  767     0761  3          THEN
;  768     0762  3              SORT_KEY [ .I,SORTKEY$W_LENGH ] =
;  769     0763  3                      ( .SORT_KEY [ .I,SORTKEY$W_LENGH ] * 2 ) - 1
;  770     0764  3
;  771     0765  3          END;
;  772     0766  3
;  773     0767  3      ! Begin the sort
;  774     0768  3      !
;  775     0769  3      SOR$BEGIN_SORT( KEY_BUFFER,          ! Key buffer address
;  776     0770  3                      LRL,                 ! Longest record length
;  777     0771  3                      SORT_OPTIONS,        ! Sort options
;  778     0772  3                      0,                   ! Input file size
;  779     0773  3                      0,                   ! Comp. routine addr.
;  780     0774  3                      0,                   ! Equal routine addr.
;  781     0775  3                      SORT_TYPE,           ! Sort type
;  782     0776  3                      WORK_FILES );        ! Number of work files
;  783     0777  3
;  784     0778  2          RETURN
;  785     0779  2
;  786     0780  1      END;


                                                .PSECT  _CONV$OWN,NOEXE,  PIC,2

                                    00078 KEY_BUFFER:
                                                .BLKB   66
                                    000BA LRL:      .BLKB   2
                                    000BC SORT_OPTIONS:
                                                .BLKB   4
                                    000C0 SORT_TYPE:
                                                .BLKB   1
                                    000C1 WORK_FILES:
                                                .BLKB   1

                                          SEGMENTS=         KEY_BUFFER
                                          SORT_KEY=         KEY_BUFFER+2


                                                .PSECT  _CONV$CODE,NOWRT,  SHR,  PIC,2

                    007C   8F   BB 00000 SET_UP_SORT:
                                                PUSHR   #^M<R2,R3,R4,R5,R6>                     ; 0647
0000'  CF     18   AE   90 00004                MOVB    S_TYPE, SORT_TYPE                       ; 0710
0000'  CF   0000G  CF   90 0000A                MOVB    CONV$GL_WORK_F, WORK_FILES              ; 0711
0000'  CF   0000G  CF   B0 00011                MOVW    CONV$GW_MAX_REC_SIZ, LRL                ; 0712
       0B     10   AB   E9 00018                BLBC    16(KEY_DESC), 1$                        ; 0716
0000'  CF 00000000G 8F   D0 0001C                MOVL    #SOR$M_STABLE, SORT_OPTIONS             ; 0718
                    04   11 00025                BRB     2$                                      ;
             0000'  CF   D4 00027 1$:            CLRL    SORT_OPTIONS                            ; 0720
0000'  CF     12   AB   9B 0002B 2$:            MOVZBW  18(KEY_DESC), SEGMENTS                  ; 0724
       53     11   AB   9A 00031                MOVZBL  17(KEY_DESC), R3                        ; 0729
                    05   12 00035                BNEQ    3$                                      ; 0731
       54          0E   D0 00037                MOVL    #14, KEY_TYPE                           ;
                    49   11 0003A                BRB     11$                                     ;
       01           53   91 0003C 3$:            CMPB    R3, #1                                  ; 0732
```

```
                                    05  12 0003F        BNEQ    4$                              ;
                      54            07  D0 00041        MOVL    #7, KEY_TYPE                    ;
                                    3F  11 00044        BRB     11$                             ;
                      03            53  91 00046 4$:    CMPB    R3, #3                          ; 0733
                                    05  12 00049        BNEQ    5$                              ;
                      54            08  D0 0004B        MOVL    #8, KEY_TYPE                    ;
                                    35  11 0004E        BRB     11$                             ;
                      06            53  91 00050 5$:    CMPB    R3, #6                          ; 0734
                                    05  12 00053        BNEQ    6$                              ;
                      54            09  D0 00055        MOVL    #9, KEY_TYPE                    ;
                                    2B  11 00058        BRB     11$                             ;
                      02            53  91 0005A 6$:    CMPB    R3, #2                          ; 0735
                                    05  12 0005D        BNEQ    7$                              ;
                      54            03  D0 0005F        MOVL    #3, KEY_TYPE                    ;
                                    21  11 00062        BRB     11$                             ;
                      04            53  91 00064 7$:    CMPB    R3, #4                          ; 0736
                                    05  12 00067        BNEQ    8$                              ;
                      54            04  D0 00069        MOVL    #4, KEY_TYPE                    ;
                                    17  11 0006C        BRB     11$                             ;
                      07            53  91 0006E 8$:    CMPB    R3, #7                          ; 0737
                                    05  12 00071        BNEQ    9$                              ;
                      54            05  D0 00073        MOVL    #5, KEY_TYPE                    ;
                                    0D  11 00076        BRB     11$                             ;
                      05            53  91 00078 9$:    CMPB    R3, #5                          ; 0738
                                    05  13 0007B        BEQL    10$                             ;
                      54            01  CE 0007D        MNEGL   #1, KEY_TYPE                    ;
                                    03  11 00080        BRB     11$                             ;
                      54            15  D0 00082 10$:   MOVL    #21, KEY_TYPE                   ;
                      55     0000'  CF  3C 00085 11$:   MOVZWL  SEGMENTS, R5                    ; 0743
                      50            01  CE 0008A        MNEGL   #1, I                           ; 0746
                                    34  11 0008D        BRB     13$                             ;
                             0000'CF40  7F 0008F 12$:   PUSHAQ  SORT_KEY[I]                     ;
                      9E            54  B0 00094        MOVW    KEY_TYPE, @(SP)+                ;
                             0000'CF40  7F 00097        PUSHAQ  SORT_KEY+2[I]                   ; 0747
                                    9E  B4 0009C        CLRW    @(SP)+                          ;
                             0000'CF40  7F 0009E        PUSHAQ  SORT_KEY+4[I]                   ; 0754
                      9E       1C AB40  B0 000A3        MOVW    28(KEY_DESC)[I], @(SP)+         ;
                      51     0000'CF40  7E 000A8        MOVAQ   SORT_KEY+6[I], R1               ; 0755
                      61       2C A04B  9B 000AE        MOVZBW  44(I)[KEY_DESC], (R1)           ;
                      05            53  91 000B3        CMPB    R3, #5                          ; 0760
                                    0B  12 000B6        BNEQ    13$                             ;
                      52            61  3C 000B8        MOVZWL  (R1), R2                        ; 0763
          56          52            01  78 000BB        ASHL    #1, R2, R6                      ;
          61          56            01  A3 000BF        SUBW3   #1, R6, (R1)                    ;
          C8          50            55  F2 000C3 13$:   AOBLSS  R5, I, 12$                      ; 0760
                             0000'  CF  9F 000C7        PUSHAB  WORK_FILES                      ; 0769
                             0000'  CF  9F 000CB        PUSHAB  SORT_TYPE                       ;
                                    7E  7C 000CF        CLRQ    -(SP)                           ;
                                    7E  D4 000D1        CLRL    -(SP)                           ;
                             0000'  CF  9F 000D3        PUSHAB  SORT_OPTIONS                    ;
                             0000'  CF  9F 000D7        PUSHAB  LRL                             ;
                             0000'  CF  9F 000DB        PUSHAB  KEY_BUFFER                      ;
             00000000G  00          08  FB 000DF        CALLS   #8, SOR$BEGIN_SORT              ;
                             007C   8F  BA 000E6        POPR    #^M<R2,R3,R4,R5,R6>             ; 0780
                                    05     000EA        RSB                                     ;

; Routine Size:  235 bytes,    Routine Base:  _CONV$CODE + 0337
```

```
;  787          0781  1
;  788          0782  0 END     ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  _CONV$PLIT                     24  NOVEC,NOWRT,  RD ,NOEXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _CONV$OWN                     194  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _CONV$GLOBAL                  152  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _CONV$CODE                   1058  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                           Library Statistics
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent     Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619       120         0       1000        00:01.8
;  _$255$DUA28:[CONV.SRC]CONVERT.L32;1  165        27        16         17        00:00.2




;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONVSORT/OBJ=OBJ$:CONVSORT MSRC$:CONVSORT/UPDATE=(ENH$:CONVSORT)

; Size:          1058 code + 370 data bytes
; Run Time:        00:22.2
; Elapsed Time:    01:16.0
; Lines/CPU Min:    2112
; Lexemes/CPU-Min: 26082
; Memory Used:  183 pages
; Compilation Complete
```

RECLDCL
LIS

RECLREC
LIS

RECLCTRL
LIS

RECLRMSIO
LIS

CONVMSG
LIS

CONVMAIN
LIS

CONVSORT
LIS

CONVVEC
LIS